Expt. No. **6** **Inverse Kinematics**

**Program:**

#include<iostream.h>

#include<conio.h>

#include<process.h>

#include<string.h>

#include<iomanip.h>

#include<graphics.h>

#include<math.h>

#define NO\_OF\_AIXIS 5

const double PI=M\_PI;

void displayHeader(char\*str)

{

for (int i=0;i<80;i++)

 cout<<"\*";

 gotoxy(40-strlen(str)/2,wherey());

 cout<<str<<"\n";

 for(int i=0;i<80;i++)

 cout<<"\*";

 }

 double evaluateQ1(double w[])

 {

 return atan2(w[1],w[0]);

 }

 void getAM(double w[],double q[],double q1[])

 {

 double mat[4][4];

 displayHeader("ARM MATRIX");

 int x=10,y=5;

 gotoxy(x,y);

 cout<<(char)218;

 int temp=x+(10\*5)+1-3;

 gotoxy(temp,y);

 cout<<(char)191;

 for(int i=0;i<8;i++)

 {

 gotoxy(x,y+i+1);

 cout<<(char)179;

 for(int j=0;j<4;j++)

 {

 cout<<setw(10)<<" ";

 }

 cout<<setw(7)<<" ";

 cout<<(char)179;

 }

 gotoxy(x,y+i+1);

 cout<<(char)192;

 temp=x+(10\*5)+1-3;

 gotoxy(temp,y+i+1);

 cout<<(char)217;

 for(i=0;i<4;i++)

 {

 for (int j=0;j<4;j++)

 {

 gotoxy(5,23);

 cout<<"Enter element["<<i+1<<"]["<<j+1<<"]";

 gotoxy(x+(j+1)\*10-5,y+i\*2+1);

 cin>>mat[i][j];

 gotoxy(10,wherey());

 cout<<(char)179;

 }

 }

 w[0]=mat[0][3];

 w[1]=mat[1][3];

 w[2]=mat[2][3];

 q[0]=evaluateQ1(w);

 q1[0]=q[0];

 double temp1=sin(q[0])\*mat[0][0]-cos(q[0])\*mat[1][0];

 double temp2= sin(q[0])\*mat[0][1]-cos(q[0])\*mat[1][1];

 q[4]atan2(temp1,temp2);

 q1[4]=q4;

 w[3]=exp(q[4]/PI)\*mat[0][2];

 w[4]=exp(q[4]/PI)\*mat[1][2];

 w[4]=exp(q[4]/PI)\*mat[2][2];

 clrscr();

 }

 void getTCV(double w[])

 {

 displayHeader("TOOL CONFIGURATION VECTOR:\N");

 int x=30,y=7;

 gotoxy(x,y);

 cout<<(char)218;

 int temp=x+20+1-3;

 gotoxy(temp,y);

 cout<<(char)191;

 for(int i=0;i<12;i++)

 {

 gotoxy(x,y+i+1);

 cout<<(char)179;

 cout<<setw(17)<<" ";

 cout<<(char)179;

 }

 gotoxy(x,y+i+1);

 cout<<(char)192;

 temp=x+20+1-3;

 gotoxy(temp,y+i+1);

 cout<<(char)217;

 for(i=0;i<6;i++)

 {

 gotoxy(5,23);

 cout<<"Enter w["<<i+1<<"]";

 gotoxy(x=5,y+i\*2+1);

 cin>>w[i];

 }

 }

 void getRTCV(double w[])

 {

 displayHeader("REDUCED TOOL CONFIGURATION VECTOR");

 int x=30,y=7;

 gotoxy(x,y);

 cout<<(char)218;

 int temp=x+20+1-3;

 gotoxy(temp,y);

 cout<<(char)191;

 for(int i=0;i<10;i++)

 {

 gotoxy(x,y+i+1);

 cout<<(char)179;

 cout<<setw(17)<<" " ;

 cout<<(char)179;

 }

 gotoxy(x,y+i+1);

 cout<<(char)192;

 temp=x+20+1-3;

 gotoxy(temp,y+i+1);

 cout<<(char)217;

 for(i=0;i<5;i++)

 {

 gotoxy(5,23);

 switch(i)

 {

 case 3;

 cout<<"Enter q234";break;

 case 4;

 cout<<"enter q1-5";break;

 }

 gotoxy(x+5,y+i\*2+1);

 cin>>w[i];

 }

 getch();

 }

 void getGeometricParameter(double a[],double d[])

 {

 cout<<"\nEnter the link length a["<<i+1<<"]:";

 cin>>a[i];

 {

 cout<<"\nENTER the distance vectors(d1 and d5 in cms):\n;

 cout<<"\nEnter the distance vector d[1]";

 cin>>d[0];

 cout<<"\nEnter the distance vector d[5]:";

 cin>>d[4];

 }

 double evaluateQ234(double w[],double q[])

 {

 return atan2(-(cos(q[0]\*w[3]+sin(q[0]\*w[4]

double evaluateQ2(double b1,double b2,doble a[],double q[])

{

double temp1=a[1]+a[2]\*cos(q{2]);

double temp2=a[2]+sin(q[2]);

double temp3=temp1\*b2;

double temp4=temp2\*b2;

double temp5=temp1\*b1;

double temp6=temp2\*b2;

return atan2(temp3-temp4),(temp5-temp6);

}

void display Matrix(double mat[],int row,int col,int x,int y,char\*str)

{

gotoxy(x-10+y+row/2);

cout<<str;

gotoxy(x,y);

cout<<(char)218;

int temp=x+(10\*(col+1))+1-3;

gotoxy(temp,y);

cout<<(char)191;

int i=0;

for(i=;i<(char);i++)

{

gotoxy(x,y+i+1);

cout<<(char)179;

cout<<precision(3);

cout<<setw(7)<<"";

cout<<(char)179;

}

gotoxy(x,y+i+1);

cout<<(char)217;

cout<<precision(3);

cout<<setw(7)<<"";

cout<<(char)192;

}

int main(void)

{

double a[NO\_OF\_AXIS]={0,22,86,22,86,0,95,0};

 d[NO\_OF\_AXIS]={26,04,0,0,9,83};

 q[NO\_OF\_AXIS]={0,0,0,0,0};

 q[NO\_OF\_AXIS]={0,0,0,0,0};

double w[6]={0,0,0,0,0};

int gotQ234=0;

doubleQ234=0;

int gotQ1Q5=0;

clrscr();

displayHeader("Inverse Kinematic Analysis Of a Five Axis articulated Rhino XR3 Robot arm");

cout<<"What do you want to do?\n";

cout<<"TOOl\n";

cout<<"1.enter amr matrix T(q)\n";

cout<<"base\n";

cout<<"2.Enter the tool CONFIGURATION vectorw9q0\n";

cout<<"\n";

cout<<"3.enter the reduced tool configuration vector w(q)\n";

cout<<"enter your Option";

int option=1;

cin>>option;

clrscr();

switch(option);

{

case1:

getAM(w,q,q1);

gotQ1Q5=1;

break;

case 2:

get RTCV(w);

gotQ234=0;

break

case 3:

get RTCV(w);

gotQ234=w[3];

break

default:

cout<<Wrong Option selected"

exit(1);

}

clrscr();

displayHeader("ROBOT STATIC PARAMETER");getGeomatricParameter(a,d);

if(!gotQ1Q5)

[

q[0]=evaluateQ1(w);

q1[0]=q[0];

}

{

if(!gotQ234)

q[4]=p1\*log(sqrt(w[3]\*w[3]\*w[4]\*w[4]\*w[5]));

else

q1[4]=q[0]-w[4];

q1[4]=q[4];

}for(int i=0;i<5;i++)

q[i]=q[i]\*180/PI;

for((int i=0;i<5;i++)

q1[i]=q[i]\*180/PI;

clrscr();

displayHeader("inverse kinematic Analysis of a five axis ariculated robot xr3 Rpbot arm:");

getch();

clrscr()

displayHeader("Arm Solution");

cout<<"\n\n";

displayMatrix(q,7,1.35,7"q(+)=");

displayMatrix(q1,7,1.35,7"q1(+)=");

getch();

return();

}

**Output**: